A Novel Methodology for Efficient Throughput Evaluation in Virtualized Routers

Jordi Mongay Batalla
Warsaw University of Technology
Nowowiejska Str. 15/19,
00-665 Warsaw, Poland
jordim@interfree.it

Miroslaw Kantor
Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg
L-2721 Luxembourg
miroslaw.kantor@uni.lu

Constandinos X. Mavromoustakis, Georgios Skourletopoulos
Department of Computer Science
University of Nicosia
Nicosia, Cyprus
mavromoustakis.c@unic.ac.cy,
skourletopoulos.g@unic.ac.cy

George Mastorakis
Department of Informatics Engineering
Technological Educational Institute of Crete
Heraklion, Crete, Greece
gmastorakis@staff.teicrete.gr

***Abstract*—This paper analyzes a novel methodology for calculating the throughput in a device, which hosts multiple virtualized network interconnect devices (i.e. virtual routers). The proposed methodology, which extends the well-known procedure (for non-virtualized IP routers) adopted from RFC 2544, considers the impact of heterogeneity of the offered load at the level of virtual routers. The utility of this methodology is demonstrated, analyzing the throughput of virtualized routers by four different virtualization platforms that use two different techniques, which are the paravirtualization (Xen and Citrix Xen) and the OS-level virtualization (Linux Containers and Jails). The results indicate that the virtualization platforms behave differently to distribution of traffic load among virtual routers. Finally, the need for the proposed methodology is motivated by performing extensive throughput tests on the aforementioned platforms at different work points of the network device (i.e. different offered traffic load distribution between virtual routers).**

Keywords—virtual router; test methodology; throughput

I. INTRODUCTION

Network virtualization brings several benefits to the Network Providers due to its flexibility, programmability, elasticity and dynamicity [1], [2]. More specifically, in the area of mobile telephony, more than 800 MVNOs (Mobile Virtual Network Operators) offer services to the end users, even though they do not own the wireless network infrastructure or they have not enough network resources [3]. When deploying a virtual infrastructure, the operator should also take into consideration the flexibility and performance of the virtualization platform, in addition to the cost [4], [5]. Providing a virtual environment usually requires some overhead, which is necessary for managing resources between network devices that share the same node (real hardware). Overhead depends on the deployed virtualization software (i.e. platform) and has a significant impact on the performance (especially on the throughput) and the packet losses of the routers.

In this context, this paper proposes a novel test methodology for finding the worst work point of the virtualized device, indicating the distribution of the offered traffic (among Virtual Routers) for which the virtualized device displays the worst performance. Furthermore, the proposed approach demonstrates that the comparison between different virtualization platforms should be based on the performance analysis at the worst work points of the devices, considering that the work point where the device will run is unknown. This fact helps to avoid inconsistencies during the comparison process, such as when a platform runs at its worst work point compared to its best one. In addition, the proposed methodology is applied to network devices that run four different software virtualization platforms, which are the (Linux) LXC, (FreeBSD) Jails, Xen and Citrix Xen. It is worthy to mention that the tests are performed for the LXC and Jails platforms. These platforms are not deeply analyzed in the literature, because they implement OS-level (Operating System-level) virtualization. This approach enables for having separate network contexts; an essential parameter for implementing independent routing functions.

II. RELATED WORK

The influence of the virtualization platform on the performance of virtualized network interconnect devices has been elaborated on several papers, adopting different comparison approaches [6]–[11]. However, these comparisons are not accurate due to the exploited methodology of benchmarking Virtual Routers (VRs), as the comparison is performed at only one work point of the virtualized device. The "work point" term refers to the distribution of offered traffic load between the virtual routers into the device. Authors in other published works take measurements only when all the virtual routers suffer equal traffic load, ignoring other work points of the device. Egi et al. [7] are the only to analyze the differences between Xen platforms with bridged and routed setup, ending up to the conclusion that the same platform has different behavior, which depends on the setup parameters, and it privileges some domains in the virtualization platform, resulting in differences regarding the throughput of the VRs. Furthermore, authors in [8] and [9] draw different conclusions regarding the performance of the Xen platform, presenting analyses once working at two different working points. Rathore et al. [8] present results by measuring the forwarding performance of a single virtual router in the presence of increasing number of VRs, which do not forward packet streams. In this case, the work point exists when one VR receives all the offered load. On the contrary, Mattos et al. [9] indicate scalability issues and provide a Xen Virtual Router Evaluation, however all VRs receive the same offered load.

III. Test Methodology Approach

The benchmarking test methodology, presented in [12], considers several measurements related to packet forwarding performance of IP routers, including throughput, latency, loss rate, back-to-back frames, system recovery and reset. The throughput is defined as the maximum offered traffic load that can be forwarded by the device with no packet loss; a fact that indicates the usability limit of a particular device. In the case of a virtualized network device, the total offered traffic load consists of the sum of the loads offered to each VR. Even though the total offered traffic load is a simple scalar value, there are many options for partitioning the load across particular VRs. The mathematical model of the presented issue considers the data set of loads offered to each of the *N* virtual routers $\boldsymbol{O} = \{O_i : i = 1, \dots, N\}$, which represents the work point of the virtualized device. Subsequently, the total load offered to the network device O_{tot} is the sum of the *N* offered loads. The maximum number *N* of virtual routers in the device depends on the device itself, the virtualization platform (capacity) as well as the aimed function of the virtualized device. For measuring the distribution of the traffic load among different VRs, the fairness index of Jain et. al [13] is exploited, which is defined as follows:

$$J = \frac{O_{tot}^2}{N \times \sum_{i=1}^{N} O_i^2} \tag{1}$$

The above fairness index is a measure of the second moment for the data set that, unlike the standard deviation, takes values from a limited range $J \in (0, 1]$. In the case of traffic load sharing among finite number of VRs, it ranges from $1/N$ (single VR receives the whole offered load and the remaining VRs are not loaded) to 1 (all VRs receive the same offered load). It is worthy to notice that other research papers focusing on virtualized network devices, usually assume that the same traffic is offered to all the VRs (i.e. $J = 1$).

A. Motivation

It is crucial to examine if two different $\boldsymbol{O}$ sets (with different distributions of traffic load among VRs) with the same O_{tot} result in the same forwarding performance. Towards arriving at a decision, experiments are performed with 12 VRs deployed on a HP ProLiant DL360G6 server, running Xen 4.1.2 with host kernel 2.6.56 (64bit).

Each VR (guest system) features 2 virtual network interfaces, while the host system maps the traffic to VR with VLAN tags. The traffic was generated by Spirent TestCenter C1 (equipped with CM-1G-D4 card), whereas the tester and the Device Under Test were connected by two 1 Gbps Ethernet links in ring topology, as proposed in [12] and shown in Fig. 1. The dotted line in the figure indicates the exemplary data path.

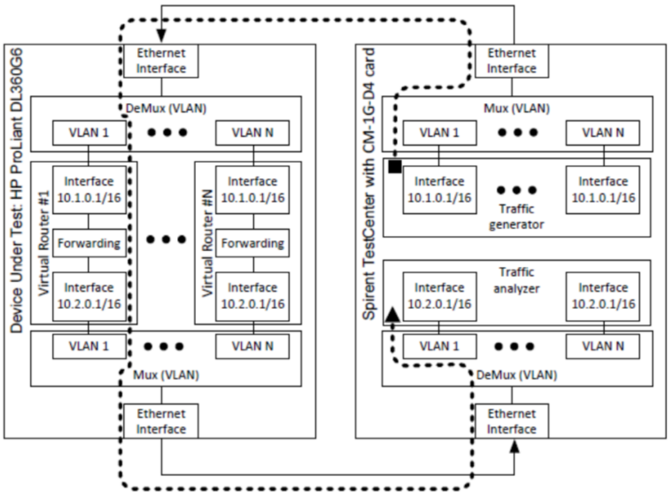

Fig. 1. Topology of the test network (including VRs)

Additionally, the Spirent TestCenter adds VLAN labels into the packets, which are used to de-multiplex packets to the appropriate VRs within the server. Each VR has its own routing table, which is examined for each packet during the forwarding process. Finally, the server multiplexes the packets into appropriate VLAN labels for forwarding the packets to the Spirent TestCenter. The tests were repeated 10 times aiming to obtain confidence intervals, which are also presented in the results. To conclude, the performance of the virtualized network device depends on the distribution of the traffic load among VRs (i.e. the work point). Consequently, the throughput measurement becomes conditional, as for $J = 1$, it is equal to 13 Mbps, while for other values of *J*, the throughput is lower. Other related work compare virtualization platforms by considering equal traffic offered to all the VRs. According to the tests, this is not necessarily the worst case work point for each platform. Towards ensuring consistence between the tests, it is taken into account that the platforms should be always compared at the worst work point. Therefore, an extension of the presented methodology in [12] is proposed related to the virtualized environment.

B. Test Methodology for Throughput

The proposed methodology aims at finding the largest amount of offered traffic that is forwarded without packet loss by the Device Under Test (DUT). Each test lasts at least 60 seconds and the Ethernet frame sizes are selected from the set $\{64, 128, 256, 512, 1024, 1280, 1518\}$. A minimum of 5 different frame sizes is tested [12]. The assumptions and operations related to the proposed methodology are as follows:

- DUT runs exactly *N* VRs and the final throughput value T^N depends on the number of running VRs.

- VRs are separated into 2 groups - k VRs are heavily loaded and $N - k$ VRs are lightly loaded. There are N possible allocations in this scheme, $k \in \{1, 2, \ldots, N\}$, and all of them are considered (all values of k).
- For each allocation, the maximum offered load O_{tot} is investigated, where all VRs have no packet loss. Part of this load (i.e. qO_{tot}) is divided equally among lightly loaded VRs for $k \neq N$, while the remaining is divided equally among heavily loaded VRs. A value $q = 0.01$ is suggested, as it ensures that the virtualization platform may handle lightly loaded VRs and, additionally, the forwarding of this amount of traffic influences slightly the heavily loaded VRs. Furthermore, the value $q = 0$ could provoke the disablement of some VRs in some platforms due to the virtual machines turnoff, as they do not use system resources during a long period of time.
- The offered load for each heavily loaded VR is given by

$$O_h = \begin{cases} \frac{(1-q)O_{tot}}{k} & k \in 1, \ldots, N-1 \\ \frac{O_{tot}}{N} & k = N \end{cases} \tag{2}$$

- The offered load for each lightly loaded VR is given by

$$O_l = \begin{cases} \frac{qO_{tot}}{N-k} & k \in 1, \ldots, N-1 \\ 0 & k = N \end{cases} \tag{3}$$

- The Jain's fairness index for given k is equal to

$$J(k) = \begin{cases} \frac{k(N-k)}{N[N(1-q)^2 - k(1-2q)]} & k \in 1, \ldots, N-1 \\ 1 & k = N \end{cases} \tag{4}$$

$$\lim_{q \to 0^+} J(k) = \frac{k}{N} \tag{5}$$

- The conditional throughput $T|_{J=J(k)}$ is equal to achieved O_{tot}.
- Regarding the final result, the throughput T^N for N VRs is defined as follows, after completing the test for all values of k:

$$T^N = \min_{k=1,\ldots,N} T|_{J=J(k)} \tag{6}$$

  which constitutes the throughput at the worst work point of the device.

It is significant to mention that through this methodology, the virtualization system does not consider hierarchy between VRs, therefore it is not important which VRs are selected as heavily loaded ones.

IV. Experimental Validation Tests

Even though the proposed methodology is relevant for both hardware and software virtualization platforms, the tests are performed only on software virtualization platforms for cost-saving purposes. Software virtualization methods involve sharing physical resources among different virtual machines, which "in isolation" make use of the aforementioned resources. Two main techniques are used in software virtualization: hardware- and OS-level. The difference lies in the placement of the virtualization layer within the device. Hardware-level virtualization locates the virtualization layer just on the top of the hardware one, whereas OS-level systems place the virtualization layer above the host OS, which, in turn, is on top of the hardware.

Furthermore, in hardware-level virtualization, each VR uses its own operating system kernel, offering advanced isolation to the VR. Instead, OS-level virtualization methods use the common kernel of the host OS, which has its system calls modified to allow multiple isolated user spaces. Therefore, OS-level methods save overhead in terms of both use of resources and operational time, however loss flexibility should communicate with the same kernel, which implies the installation of the same OS in all the VRs. Solaris is the only to have partially gone beyond this restriction by providing a container, where it is possible to install Linux 2.4-based OS [14].

Xen (also Citrix Xen) is based on paravirtualization, which is a variation of hardware-level virtualization. It allows the communication between the guest OS and the hypervisor by running modified guest kernel code with nonvirtualizable instructions replaced with calls to the hypervisor. On the other hand, solutions based on full virtualization allow to run unmodified guest OS by using hardware extensions, enabling hypervisor to intercept nonvirtualizable instructions. In addition, FreeBSD Jails and Linux LXC follow OS-level virtualization, enabling multiple isolated user spaces to run application instances. The host kernel ensures isolation and impact limitation between different application activities. The applications in the VR use the normal system call interface, resulting in a reduction of the necessary overhead for managing virtualization.

The virtualization method affects significantly the I/O devices' performance, as it has been widely studied for both hardware and software virtualization [15]. For example, access to the hardware is limited to the host kernel at hardware-level, receiving requests from the virtual Network Interface Controller (vNIC) driver in the guest kernel. Data have to be copied between the guest and host kernel, increasing the operational time of forwarding the packet. On the contrary, OS-level virtualization moves virtual interfaces into the VR. The kernel is responsible for keeping track of the owner regarding each interface, which results in no additional data copying requirements. Therefore, OS-level virtualization is assumed to be more effective in forwarding packets and the results of the performed tests confirm that assumption.

Towards virtualizing the IP routers in the device, the kernel should have built-in support for providing separate network context for each VR. Due to this requirement, some OS-level virtualization systems, such as the iCore Virtual Accounts, cannot be used for virtualizing routers. This is the major reason for which OS-level virtualization has been uniquely used in virtual hosting environment for many years. Nowadays, many OS-level virtualization systems, such as Jails and LXC, guarantee separate network context, making feasible that each virtual router has its own routing table. Moreover, each VR contains its own ARP table, which allows IP address space isolation, if needed. Besides, the IP address space isolation requires a link layer multiplexing mechanism (i.e. with VLAN tagging), whenever one physical interface is shared among multiple virtual interfaces.

Finally, there are also some minor differences between OS-level virtualization methods. For instance, LXC allows to assign the same name to two or more virtual interfaces located in different VRs, while Jails does not. However, since these issues are only related to network device configuration, they do not affect the system's functionality and performance.

A. Finding the Worst Work Point of the DUT

Towards presenting the test results, the proposed methodology was exploited to analyze the Xen version 4.1.2 with host kernel 2.6.56 (64 bit) and Linux Containers (LXC) version 0.7.6 with kernel 2.6.56 (64 bit), which were installed on the hardware. The selection was intentional, since two platforms, whose behaviors were different with regard to sharing load between VRs (i.e. worst work points are different), were considered. Additionally, the results indicate the usefulness of the proposed methodology, as it enables to adequately compare virtualization systems.

The DUT is connected to the tester by two 1 Gbps Ethernet links in ring topology, as indicated in Fig. 1, and a number (arbitrarily limited to 12) of virtual machines was installed with functionality of IP router (VRs), which was configured using standard routing mechanisms from OS kernel. Each VR owns its control and data plane (also routing table), however the unique functionality that is analyzed in the tests is packet forwarding. Furthermore, each generated traffic stream has a constant bit rate profile and it uses UDP transport protocol. Following the requirements in [12], five frame sizes are considered - 64, 128, 512, 1024, 1518 bytes – and the tests were repeated 10 times, aiming to calculate 95% of the confidence intervals.

The throughput results (for different values of Jain's fairness index) for VRs virtualized on the LXC platform are presented in Fig. 2, while the throughput results of VRs virtualized on the Xen platform are shown in Fig. 3. Different curves present results for configuration of 1, 6 and 12 VRs with frame size of 64 bytes.

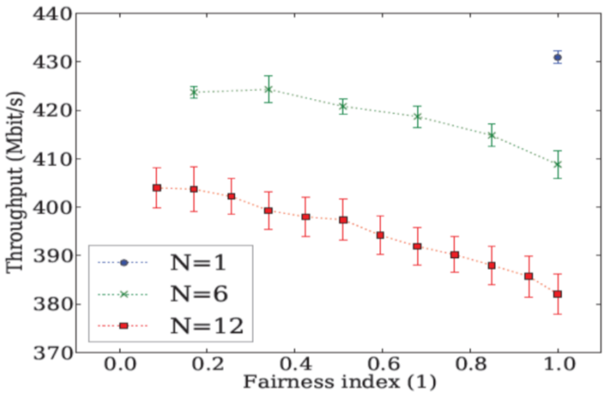

Fig. 2. Throughput as a function of Jain's fairness index for LXC virtualization platform

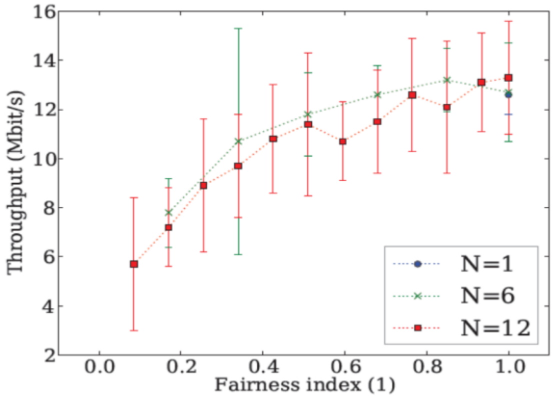

Fig. 3. Throughput as a function of Jain's fairness index for Xen virtualization platform

The measurements indicate differences (more than one order of magnitude) between the two different virtualization platforms due to the smaller overhead for OS-level techniques compared to the hardware-virtualization ones. The most interesting result is that LXC finds the minimum throughput for a fair repartition of the load between all the VRs ($J = 1$), whereas Xen finds the minimum throughput at the work point, once maximum unfairness is achieved ($J \approx 1/N$). The complexity of the virtualization platforms makes difficult to reach a conclusion about this duality, however an explanation can be given, assuming a high simplification of the virtualization system to one single scheduler that shares the access to the system resources between the VRs. More specifically, whenever the scheduler is non-work-conserving, the worst case would be closer to the work point, where one VR carries most of the traffic, i.e. $J \approx 1/N$, which corresponds to the Xen case. On the contrary, a work-conserving scheduler is linked to a worst case for $J = 1$, as it occurs for LXC. The overall performance of the non-work-conserving scheduler is usually less efficient. Since LXC and Xen exhibit worst case performance for different values of J, it is pointless to compare them in the same work point and, thus, the comparison should be provided between the worst work points for each platform.

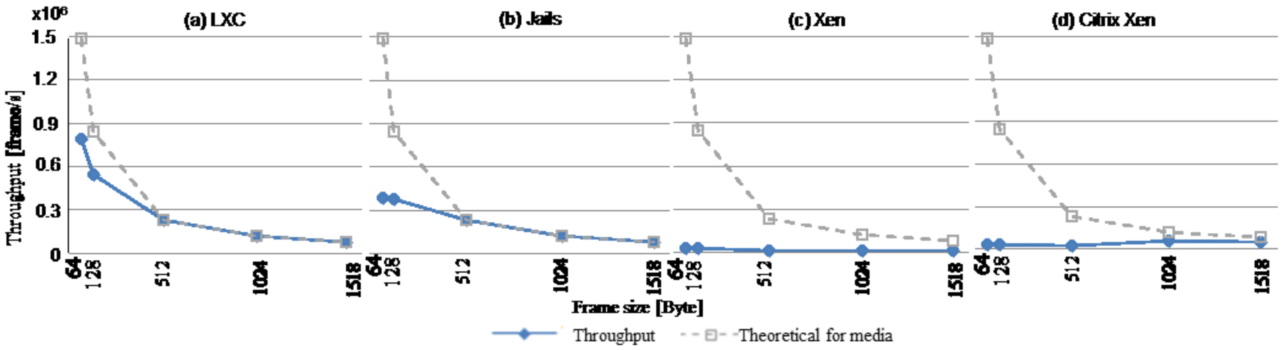

Fig. 4. Throughput in different virtualization systems for different frame sizes

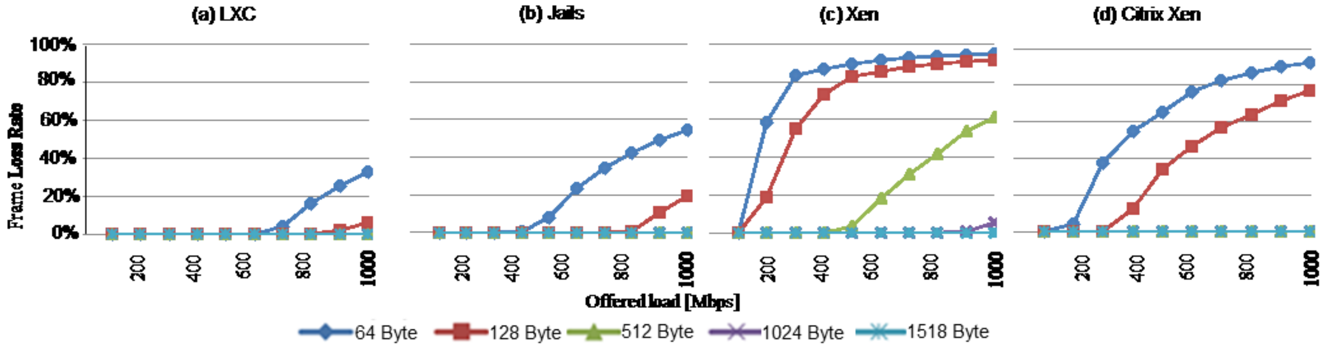

Fig. 5. Frame Loss Rate in different virtualization systems for different frame sizes

According to the results, it is witnessed that the confidence interval increases with *N*, namely the device suffers more fluctuations when more VRs are installed. The throughput is inversely proportional to *N*, indicating that the overhead in the system increases with the number of the installed VRs. This effect is also observed for VRs in other works [16]. The values of throughput define the minimum throughput that the device reaches for any distribution of offered load among VRs, enabling the performance comparison for different virtualization platforms. The results regarding LXC are always equal to 1 Gbps for the 512, 1024 and 1528 byte frame sizes, as VRs do not drop frames. Finally, test results for Jails and Citrix Xen are not presented due to limited space. However, Jails behaves similarly to LXC and finds the worst work point for $J = J(N) = 1$, whereas Citrix Xen finds the worst work point for $J = J(1) \approx 1/N$, as the one occurred for Xen.

B. Packet Forwarding Performance

A performance comparison between OS-level and hardware-level virtualization platforms is performed, in the case of one installed VR in the platforms as well as when multiple VRs compete for the resources. In the latter case, the systems work at two work points, which are the $J = J(N) = 1$ (worst work point for LXC and Jails) and $J = J(1) \approx 1/N$ (worst work point for Xen and Citrix). The test scenario and conditions are the ones proposed in Fig. 1. The Citrix platform is Citrix Xen Server 5.6.8 with host kernel 2.6.56, while FreeBSD Jails uses kernel 8.3-Release (recompiled with option VIMAGE). Xen and LXC platforms are the same as in previous sections and the tester sends streams of frames to each VR. Each stream has Constant Bit Rate and uses UDP as the transport protocol. The frames' sizes are equal to 64, 128, 512, 1024 and 1518 bytes. Additionally, the tests were repeated 10 times in order to calculate the confidence intervals. All the results have confidence intervals smaller than 15% of the mean values at the 95% of the confidence level, while the confidence intervals are not presented in the figures in order to be readable.

The scope of the initial tests is to compare the different virtualization platforms, when there is no rivalry for the resources due to the fact that only one single VR competes for the CPU. The throughput of the different virtualization systems for different frame sizes is illustrated in Fig. 4 and the throughput is compared to the theoretical load for the media (1 Gbps). On the contrary, the loss rate, when increase on the offered load occurs, is shown in Fig. 5. Both figures are strongly associated, as the throughput is the maximum offered load forwarded by the device without any losses and more offered traffic than throughput causes more losses. Moreover, the magnitude of these losses is presented in Fig.5. The loss ratio is often very low and it is not possible to evaluate the values in the figure. For example, in LXC, the throughput for 128 byte frame size is 0.55x106 frame/s (i.e. 560 Mbps), as shown in Fig. 4 (a), however the losses for the 128 byte stream are only noticeable at the range of 900 Mbps, as presented in Fig. 5 (a). The results show that the overheads for Xen and Citrix are higher than for Jails and LXC, as expected. LXC performs better than Jails; possibly due to the better implementation of the NIC drivers in Linux. The losses in Xen present a sharper deviation than the Citrix Xen ones, meaning that the commercial version of Xen (Citrix) outperforms compared to the freeware Xen version.

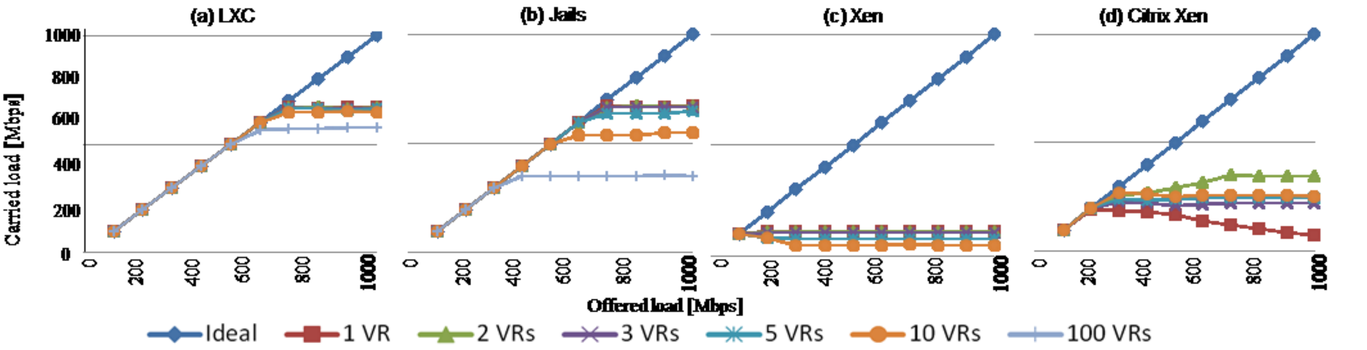

Fig. 6. Total Carried Load vs. Total Offered Load in different virtualization systems for different number of VRs (Work point: $\boldsymbol{J}=\boldsymbol{J}(\boldsymbol{N})=\mathbf{1}$, Frame Size: 64 bytes)

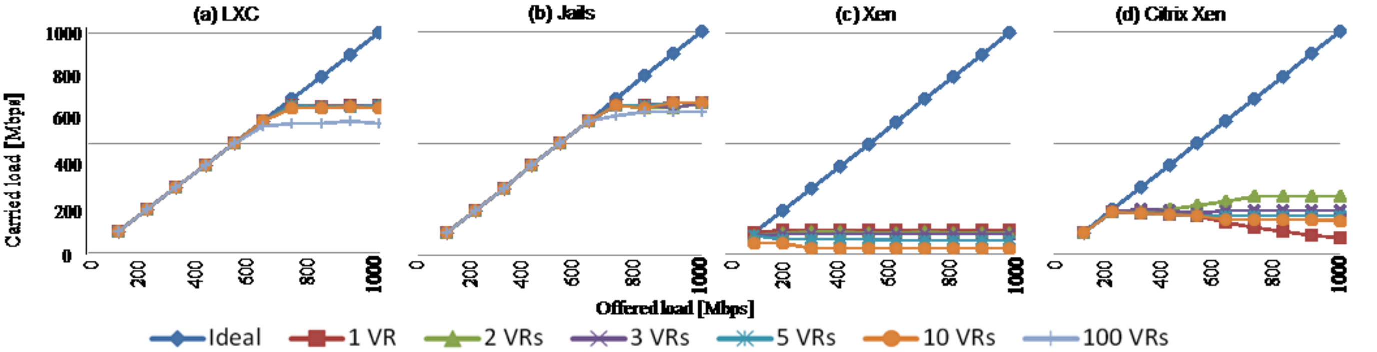

Fig. 7. Total Carried Load vs. Total Offered Load in different virtualization systems for different number of VRs (Work point: $\boldsymbol{J}=\boldsymbol{J}(\mathbf{1})\approx\mathbf{1}/\boldsymbol{N}$, Frame Size: 64 bytes)

Regarding the next tests, up to 100 VRs were tested for Jails and LXC, but only up to 10 VRs with respect to Xen and Citrix Xen due to memory capacity limitations of HP ProLiant device. The relation between offered and carried load in LXC, Jails, Xen and Citrix Xen is presented in Fig. 6 and 7 for 64-byte frame size, while Fig. 8 and 9 illustrate the same relation for 128-byte frame size. Results for other frame sizes are not provided due to limited space, however the behavior is similar to the presented ones. Towards presenting the performance comparison analysis between the platforms, the results for $J=J(N)=1$ (Fig. 6 and 8) and $J=J(1)\approx 1/N$ (Fig. 7 and 9) are provided. Narrowing in, the throughput value for the given number of VRs meets the higher value of offered load, for which the offered load matches to the carried one. Higher values of offered load cause losses, which can be easily calculated from the values of offered and carried load. Henceforth, the results demonstrate that OS-level virtualization methods are more effective than the hardware-level ones, conforming to the statement that OS-level overhead is smaller than the hardware-level virtualization one, especially for the access to the Network Interface Controller. On the other hand, the results demonstrate that the performance decreases when the number of VRs increases (due to the increase of overhead), i.e. $\forall_{k<N}T^k \geq T^N$. This characteristic is not maintained in the Citrix Xen platform, where the behavior is more unpredictable.

The comparison between virtualization platforms should be performed at the worst work point, so the proper figures to be compared are Fig. 6 (a) and (b) and Fig. 7 (c) and (d), respectively. Comparing the platforms at other work points could be misleading, such as in Fig. 7, where Jails is considered to be better than LXC; a fact that is not true for other work points different to $J=J(1)$. Furthermore, the carried load is not a strictly monotonic function of the offered load as presented in Fig. 6 (c) and (d) and Fig. 9 (c) and (d). This fact strengthens the approach that Xen has a non-work-conserving behavior for handling the resource sharing, interpreting that it is necessary to accurately condition the offered traffic in order to avoid more traffic arriving to the network device than throughput, since the performance of the device decreases in these conditions (i.e. overload conditions). The system's performance with Citrix Xen is not a monotonic function of the number of VRs, as observed in Fig. 6 (d) and Fig. 9 (d). For instance, the loss rate for 3 VRs is higher than for 1, however lower than for 2. Therefore, the way of optimizing the performance enables the access to the CPU by a number of VRs. As a result, Citrix Xen privileges some domains, since it occurs for bridged and routed setup of Xen [7]. One version of Citrix Xen is tested and, hence, it is not possible to reach conclusions about the Citrix Xen platform as a whole.

V. Conclusion

A novel methodology for virtualized network interconnect devices is proposed, illustrating the behavior of the virtualization platforms. The throughput depends also on the interaction between the VRs within the network device. Furthermore, it focuses on testing the throughput when the total

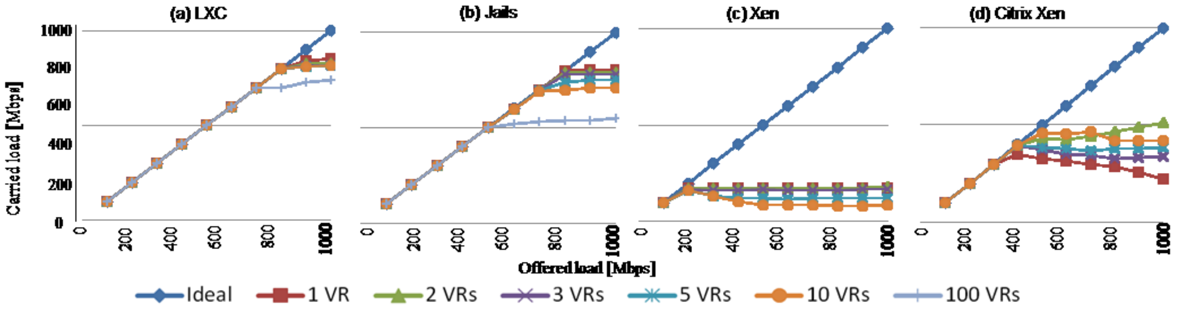

Fig. 8. Total Carried Load vs. Total Offered Load in different virtualization systems for different number of VRs (Work point: $\boldsymbol{J = J(N) = 1}$, Frame Size: 128 bytes)

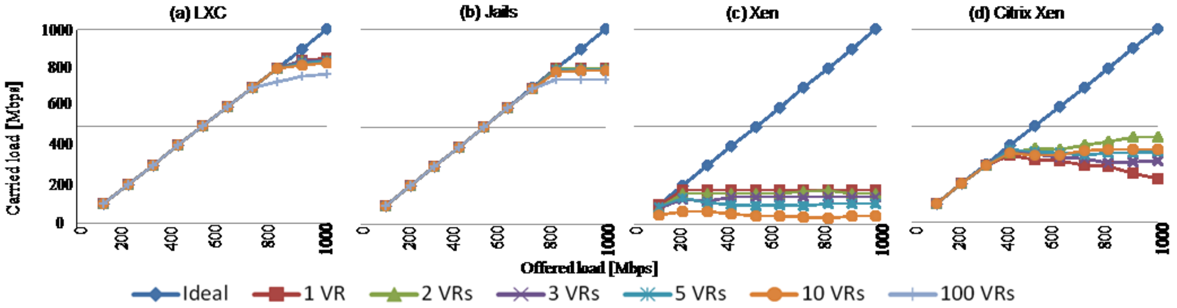

Fig. 9. Total Carried Load vs. Total Offered Load in different virtualization systems for different number of VRs (Work point: $\boldsymbol{J = J(1) \approx 1/N}$, Frame Size: 128 bytes)

load is shared out between VRs with different combinations of load and different values of the fairness index. The advantage of OS-level virtualization is also indicated from the forwarding viewpoint, once all routers can use the same software.

ACKNOWLEDGEMENT

This work was undertaken under the Pollux II IDSECOM project supported by the National Research Fund Luxembourg and the National Centre for Research and Development in Poland. The authors would like to thank Stanisław Janikowski and Jarosław Śliwiński for the support in this work.

REFERENCES

[1] M. Melo, J. Carapinha, S. Sargento, U. Killat, and A. Timm-Giel, “A Re-optimization Approach for Virtual Network Embedding,” in *Mobile Networks and Management*, vol. 58, Springer Berlin Heidelberg, 2013, pp. 271–283.

[2] M. Melo, J. Carapinha, and S. Sargento, “Network Virtualization: A step closer for seamless resource mobility,” in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, Ghent, 2013, pp. 692–695.

[3] Federal Communications Commission, “Annual report and analysis of competitive market conditions with respect to mobile wireless, including commercial mobile services,” *WT Docket*, no. 10–133, 2011.

[4] S. Zhang, Z. Qian, J. Wu, and S. Lu, “An opportunistic resource sharing and topology-aware mapping framework for virtual networks,” in *2012 Proceedings IEEE INFOCOM*, Orlando, FL, 2012, pp. 2408–2416.

[5] J. Nogueira, M. Melo, J. Carapinha, and S. Sargento, “A platform for Operator-driven Network Virtualization,” in *2011 IEEE International Conference on Computer as a Tool (EUROCON)*, Lisbon, 2011, pp. 1–4.

[6] B. Parreira, M. Melo, J. Soares, J. Carapinha, R. Monteiro, and S. Sargento, “Network Virtualization - A Virtual Router Performance Evaluation,” in *CRC 2012: 12ª Conferência sobre Redes de Computadores*, 2013, pp. 23–29.

[7] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, L. Mathy, and P. Papadimitriou, “A platform for high performance and flexible virtual routers on commodity hardware,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 1, pp. 127–128, Jan. 2010.

[8] M. S. Rathore, M. Hidell, and P. Sjödin, “Data plane optimization in open virtual routers,” in *NETWORKING 2011*, vol. 6640, Springer Berlin Heidelberg, 2011, pp. 379–392.

[9] D. M. F. Mattos, C. Fragni, M. D. D. Moreira, L. H. G. Ferraz, L. H. M. K. Costa, and O. C. M. B. Duarte, “Evaluating virtual router performance for the future internet,” Universidade Federal do Rio de Janeiro, Grupo de Teleinformatica e Automacao - PEE/COPPE - DEL/POLI, Rio de Janeiro, Brazil, 2010.

[10] D. M. Mattos, L. H. G. Ferraz, L. H. M. Costa, and O. C. Duarte, “Evaluating virtual router performance for a pluralist future Internet,” in *Proceedings of the 3rd International Conference on Information and Communication Systems (ICICS ’12)*, 2012, p. 4.

[11] M. J. Schultz and P. Crowley, “Performance analysis of packet capture methods in a 10 Gbps virtualized environment,” in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, Munich, 2012, pp. 1–8.

[12] S. Bradner and J. McQuaid, “Benchmarking methodology for network interconnect devices,” *RFC*, 1999.

[13] R. Jain, A. Durresi, and G. Babic, “Throughput fairness index: An explanation,” The Ohio State University, Department of CIS, Columbus, OH, 1999.

[14] D. Sinclair, E. Adap, and F. Sing-yan, “Linux to Solaris. Administrators Guide,” Treklogic Inc., Documents: ISO 9001:2000, Jan. 2007.

[15] M. Bourguiba, K. Haddadou, I. El Korbi, and G. Pujolle, “A container-based I/O for virtual routers: Experimental and analytical evaluations,” in *2011 IEEE International Conference on Communications (ICC)*, Kyoto, 2011, pp. 1–6.

[16] N. Egi, A. Greenhalgh, M. Handley, M. Hoerdt, F. Huici, and L. Mathy, “Towards high performance virtual routers on commodity hardware,” in *Proceedings of the 2008 ACM CoNEXT Conference (CoNEXT ’08)*, 2008, p. 20.